

**ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
“УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ”
ІНЖЕНЕРНО-ТЕХНІЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА КОМП’ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ**

ЗАТВЕРДЖУЮ
Декан інженерно-технічного
факультету
Юліана ГОЛИК
_____ 2025 р.



**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ЛОГІЧНЕ І ФУНКЦІОНАЛЬНЕ ПРОГРАМУВАННЯ**

Рівень вищої освіти – перший (бакалавр)
Галузь знань – 12 Інформаційні технології
Спеціальність – 123 Комп’ютерна інженерія
Освітня програма – Комп’ютерні системи та мережі
Статус дисципліни – вибіркова
Мова навчання – українська

Ужгород – 2025


Робоча програма навчальної дисципліни «Логічне і функціональне програмування» для здобувачів вищої освіти галузі знань 12 – «Інформаційні технології» спеціальності 123 – «Комп'ютерна інженерія» освітньої програми «Комп'ютерні системи та мережі» – 16 с.

Розробники:

Балога С.І., доцент, канд. фіз.-мат. наук, доцент кафедри комп'ютерних систем та мереж

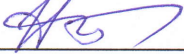
Робочу програму розглянуто та затверджено на засіданні кафедри комп'ютерних систем та мереж

протокол 13 від «25» червня 2025 р.

Завідувач кафедри  доц. Петро ГОРВАТ
(підпис) (прізвище та ініціали)

Схвалено науково-методичною комісією інженерно-технічного факультету

протокол 6 від «27» червня 2025 р.

Голова науково-методичної комісії  доц. Володимир ЦИГИКА
(підпис) (прізвище та ініціали)

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

| Найменування показників | Розподіл годин за навчальним планом |
|--|-------------------------------------|
| | денна форма навчання |
| Кількість кредитів ЄКТС – 3 | Рік підготовки: |
| Загальна кількість годин – 120 | 3-й |
| Кількість модулів – 2 | Семестр |
| | 6-й |
| Тижневих годин для денної форми навчання: аудиторних – 4,3 години самостійної роботи студента – 4,3 години | Лекції |
| | 30 год |
| | Практичні (семінарські) |
| | - |
| Вид підсумкового контролю: залік | Лабораторні |
| | 30 год |
| Форма підсумкового контролю: усна | Самостійна робота |
| | 60 год |

2. МЕТА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мета вивчення навчальної дисципліни «Логічне та функціональне програмування» – вивчення математичних основ, засад, прийомів і способів логічного та функціонального програмування; практичне засвоєння засобів функціонального та логічного програмування для розв'язування практичних та наукових задач. Оволодіння базовими можливостями мов логічного та функціонального програмування на прикладі мов Prolog і Lisp.

Завдання дисципліни – формувати теоретичні знання та практичні навички у майбутніх фахівців відповідно до поставленої мети.

Відповідно до освітньої програми, вивчення дисципліни сприяє формуванню у здобувачів вищої освіти таких компетентностей:

- інтегральна (здатність розв'язувати складні спеціалізовані задачі та практичні проблеми під час професійної діяльності у комп'ютерній галузі або навчання, що передбачає застосування теорій та методів комп'ютерної інженерії і характеризується комплексністю та невизначеністю умов);
- загальні (ЗК1 здатність до абстрактного мислення, аналізу і синтезу, ЗК2 здатність вчитися і оволодівати сучасними знаннями; ЗК3 здатність застосовувати знання у практичних ситуаціях, ЗК7 вміння виявляти, ставити та вирішувати проблеми);
- фахові (ФК2 здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення, ФК3 здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж, ФК11 здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів, ФК15 здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення).

3. ПЕРЕДУМОВИ ДЛЯ ВИВЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Передумовами вивчення навчальної дисципліни «Логічне та функціональне програмування» є опанування курсів «Дискретна математика», «Програмування», «Структури даних та алгоритми» освітньої програми «Комп'ютерні системи та мережі».

4. ОЧІКУВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ

Відповідно до освітньої програми «Комп'ютерні системи та мережі», вивчення навчальної дисципліни повинно забезпечити досягнення здобувачами вищої освіти таких програмних результатів навчання (ПРН):

| Програмні результати навчання | Шифр ПРН |
|--|----------|
| Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж. | ПРН1 |
| Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах. | ПРН2 |

| | |
|---|-------|
| Вміти системно мислити та застосовувати творчі здібності до формування нових ідей | ПРН8 |
| Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання. | ПРН10 |
| Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії. | ПРН11 |
| Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення | ПРН16 |

Очікувані результати навчання, які повинні бути досягнуті здобувачами освіти після опанування навчальної дисципліни «Логічне та функціональне програмування»:

| Очікувані результати навчання з дисципліни | Шифр ПРН |
|--|-------------------------|
| Знати і розуміти основні поняття інформатики, математики та програмування. Вміння шукати необхідну інформації для вирішення отриманого завдання на лабораторних заняттях, для розв'язування додаткових завдань, аналізувати та оцінювати її. Знати основні методи та прийоми, які використовуються у функціональному та логічному програмуванні, засоби для реалізації програм на мовах високого рівня Lisp та Prolog. | ПРН1, ПРН2, ПРН8, |
| Уміти складати програми на мовах логічного та функціонального програмування, застосовувати функціональні можливості Prolog та Lisp для обробки рекурсивних структур даних, розв'язування задач. | ПРН10 |
| Розробляти програмне забезпечення на основі вивченого теоретичного матеріалу для отриманого завдання. | ПРН11 |
| Презентувати, обговорювати та захищати власний вибір та засоби реалізації проектів виконаних в усній чи електронній формах в межах лабораторних робіт та індивідуальних завдань, підтверджувати правильність рішень відповідними тестовими прикладами. | ПРН16 |

5. ЗАСОБИ ДІАГНОСТИКИ ТА КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Робоча програма з дисципліни “Логічне та функціональне програмування”, що читається на третьому курсі ІТФ спеціальності комп'ютерна інженерія, має два модулі.

Перший і другий модулі складаються з п'яти і семи тем відповідно.

Використовуються методи усного та письмового контролю. Поточний контроль передбачає: опитування студентів під час захисту лабораторних робіт та опитування на лекціях; контрольні роботи, індивідуальні та самостійні завдання. Підсумковий контроль передбачає залік.

Для контролю знань розроблено: перелік теоретичних питань та типових завдань, (наведено в додатку); завдання для самостійної роботи, зі змістом яких студенти ознайомлюються на початку семестру.

Оцінка ECTS, яку студент отримує після вивчення кредитного модуля дисципліни, визначається відповідно до рейтингу студента. Рейтинг студента з кредитного модуля складається з балів, що він отримує протягом семестру за такі види робіт:

1. Модульна контрольна робота (МКР) тривалістю по 2 акад. години. Максимальна кількість балів за МКР – 50 балів.

2. Виконання лабораторних робіт.

Протягом вивчення дисципліни студенти виконують по 3 лабораторні роботи у кожному з модулів (максимальна кількість балів – 40 за кожен із модулів).

Бали із індивідуальної та самостійної роботи студентів нараховуються за: підготовку рефератів, модернізацію завдань, за творчий підхід до виконання завдань, виконання завдань із удосконалення дидактичних матеріалів з дисципліни: 0-10 балів за кожен модуль.

Кожний модуль оцінюється максимально в 100 балів. В кінці дисципліни виводиться рейтинговий бал, який визначається як середнє арифметичне балів з двох модулів.

Необхідною умовою допуску до заліку є відсутність заборгованостей з лабораторних робіт та зарахування контрольних робіт.

Розподіл балів, які отримують студенти за модуль приведені в таблицях.

Розподіл балів, які отримують здобувачі вищої освіти (модуль 1)

| Поточне тестування (лабораторні роботи) | | | | | Самостійна робота | Письмова контрольна робота | Сума |
|---|----|----|----|----|-------------------|----------------------------|------|
| T1 | T2 | T3 | T4 | T5 | 10 | 50 | 100 |
| 5 | 10 | 15 | 5 | 5 | | | |

Розподіл балів, які отримують здобувачі вищої освіти (модуль 2)

| Поточне тестування (лабораторні роботи) | | | | | | | Самостійна робота | Письмова контрольна робота | Сума |
|---|----|----|----|----|----|----|-------------------|----------------------------|------|
| T1 | T2 | T3 | T4 | T5 | T6 | T7 | 10 | 50 | 100 |
| 3 | 8 | 12 | 3 | 2 | 8 | 2 | | | |

Оцінювання окремих видів навчальної роботи з дисципліни

| Вид діяльності здобувача вищої освіти | Модуль 1 | | Модуль 2 | |
|---|-----------|---------------------------------------|-----------|---------------------------------------|
| | Кількість | Максимальна кількість балів (сумарна) | Кількість | Максимальна кількість балів (сумарна) |
| Лабораторні заняття (допуск, виконання та захист) | 3 | 40 | 2 | 40 |
| Самостійна робота | 1 | 10 | 1 | 10 |
| Модульна контрольна робота | 1 | 50 | 1 | 50 |
| Разом | | 100 | | 100 |

Модульна контрольна робота містить п'ять завдань. Перші два завдання включають теоретичний матеріал, наступні 3 завдання – це практична частина, за допомогою якої можна дізнатись про засвоєння матеріалу. В залежності від відповіді студента на питання, вони оцінюються від 0 до 10 балів за кожне питання. Максимальна кількість балів за модульну контрольну роботу дорівнює 50 балів.

До складання екзамену допускаються лише студенти, які мають рейтинговий бал не менше 35 і, яким зараховано всі лабораторні роботи за цей семестр.

Здобувач вищої освіти, підсумкова модульна оцінка якого складає від 0 до 34 балів, зобов'язаний покращити її до початку підсумкового семестрового контролю. В протилежному випадку, здобувач не допускається до екзамену і у нього виникає академічна заборгованість.

Залік з навчальної дисципліни студент може не скласти, якщо він склав усі модулі та його влаштовує рейтингова оцінка. Студенти, які мають рейтинговий бал від 35 до 59 іспит складають обов'язково. Студент може підвищити на екзамені чи заліку оцінку, при цьому рейтингова оцінка не може бути зменшена.

Залік проводиться в усній формі. На екзамен виносяться теоретичні та практичні завдання в обсязі навчального матеріалу за семестр. Оцінювання результатів навчання на заліку здійснюється за 100-бальною шкалою. Оцінка за екзамен вноситься у відомість обліку успішності.

За результатами виконання студентом навчальної програми впродовж семестру рекомендується виставляти заліки та екзамени без додаткового опитування за такою шкалою:

| Сумарні бали | Оцінка ECTS | Екзамен (диф. залік) | Залік |
|--------------|-------------|--|--|
| 90 – 100 | A | Відмінно | Зараховано |
| 82 – 89 | B | Добре | |
| 74 – 81 | C | | |
| 64 – 73 | D | Задовільно | |
| 60 – 63 | E | | |
| 35 – 59 | FX | Незадовільно з можливістю повторного складання | Незараховано з можливістю повторного складання |
| 1 – 34 | F | Незадовільно з обов'язковим повторним вивченням дисципліни | Незараховано з обов'язковим повторним вивченням дисципліни |

6. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

6.1. Зміст навчальної дисципліни

Змістовний модуль 1. Основні принципи і методи побудови логічних програм.

Тема 1. Предмет дисципліни та її основні завдання, зв'язок з іншими дисциплінами спеціальності. Програмування за допомогою функцій та процедур. Пролог – як мова розроблення систем штучного інтелекту. Математична основа мови логічного програмування. Поняття термінів та їх уніфікація; правила резолюцій. Факт, відношення, правило. Етапи програмування на Пролозі. Алфавіт мови Пролог. Види термів, константи, змінні та структури. Оператори, властивості операторів. Предикати в Пролозі, цілі, кон'юнкція цілей.

Тема 2. Основні елементи мови Пролог. Управління програмою. Арифметика мови Пролог. Арифметичні обчислення. Діаграми успішного доведення цільових тверджень. Декларативний і процедурний зміст програм. Механізм повернення та його застосування для доведення істинності цільових тверджень. Приклади програм з використанням механізму повернення. Механізм відсікання і способи його застосування у мові Пролог. Проблеми, пов'язані з використанням цього механізму.

Тема 3. Рекурсивне представлення даних і програм. Робота зі списками. Процедури опрацювання списків. Приклади програм з рекурсивними означеннями. Вбудовані предикати для роботи зі списками. Бінарні дерева пошуку. Предикати введення-виведення; ідентифікація типів; синтез і декомпозиція термінів; налаштування програм на Пролозі.

Тема 4. Операції з базою даних. Предикати для роботи з базою даних.

Тема 5. Технології побудови експертних систем. Експертні системи, що ґрунтуються на правилах. Експертні системи, що ґрунтуються на законах логіки.

Змістовний модуль 2. Основи функціонального програмування.

Тема 1. Основи функціонального програмування. Порівняння можливостей мов функціонального та логічного програмування з іншими мовами високого рівня. Лямбда обчислення. Величини. Функції. Програмування за допомогою функцій та процедур. Порівняння можливостей мов функціонального та логічного програмування з іншими мовами високого рівня. Функціональні мови програмування.

Тема 2. Мова функціонального програмування Lisp. Основні поняття. Символьні дані: атоми, константи, списки, символьні вирази та представлення даних.

Тема 3. Побудова рекурсивних функцій: означення рекурсії, розгляд випадків, вибір підфункцій, підходи до побудови, запис S- виразу.

Тема 4. Спискова форма представлення даних у функціональному програмуванні. Принципи опрацювання списків. Вкладання списків на довільну глибину. Спрощена форма запису.

Тема 5. Суть крапкової форми подання виразів. Використання його до побудови функціональних програм. Приклади побудови функціональних програм.

Тема 6. Принципи структурованої побудови програм та їх структурованого відлагоджування. Побудова програм аналізу розмірностей математичних формул – як приклад структурованої програми. Два підходи до подання розмірностей.

Тема 7. Додаткові можливості функціональної мови Lisp. Функції вищих порядків. Локальні форми, лямбда-вирази, параметри нагромадження та їх застосування до побудови функціональних програм. Означення та побудова функцій вищих порядків, їх функціональне означення та запис S-виразів.

6.2. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

| Назви змістових модулів і тем | Кількість годин | | | | | |
|---|-----------------|--------------|-----------|------------|-------------|------------|
| | Денна форма | | | | | |
| | Усього | у тому числі | | | | |
| | | Лекції | практичні | лабораторн | індивідуаль | самостійна |
| | | | | на робота | робота | |
| <i>1</i> | 2 | 3 | 4 | 5 | 6 | 7 |
| Модуль 1 | | | | | | |
| Змістовний модуль 1. Основні принципи і методи побудови логічних програм. | | | | | | |
| Тема 1. Предмет дисципліни та її основні завдання, зв'язок з іншими дисциплінами спеціальності. Загальні відомості про логічне програмування. | 2 | 2 | | | | |
| Тема 2. Основні елементи мови Пролог. Управління програмою. Арифметика мови Пролог. | 15 | 3 | | 4 | | 8 |
| Тема 3. Рекурсивне представлення даних і програм. | 30 | 8 | | 10 | | 12 |
| Тема 4. Операції з базою даних. | 1 | 1 | | | | |
| Тема 5. Технології побудови експертних систем. | 12 | 2 | | | | 10 |
| Разом за модуль 1 | 60 | 16 | | 14 | | 30 |
| Модуль 2 | | | | | | |
| Змістовний модуль 2. Основи функціонального програмування. | | | | | | |
| Тема 1. Основи функціонального програмування. | 1 | 1 | | | | |
| Тема 2. Мова функціонального програмування Lisp. Основні поняття. | 5 | 1 | | 4 | | |
| Тема 3. Побудова рекурсивних функцій: означення рекурсії, розгляд випадків, вибір підфункцій, підходи до побудови. | 15 | 4 | | 4 | | 7 |
| Тема 4. Спискова форма представлення даних у функціональному програмуванні. | 13 | 2 | | 4 | | 7 |
| Тема 5. Суть крапкової форми подання виразів. | 8 | 2 | | | | 6 |
| Тема 6. Принципи структурованої побудови програм та їх структурованого відлагоджування. | 6 | 2 | | 4 | | |

| | | | | | | |
|--|------------|-----------|--|-----------|--|-----------|
| Тема 7. Додаткові можливості функціональної мови Lisp. Функції вищих порядків. | 12 | 2 | | | | 10 |
| Разом за модуль 2 | 60 | 14 | | 16 | | 30 |
| Разом за семестр | 120 | 30 | | 30 | | 60 |

6.3. Теми лабораторних занять

| № з/п | Назва теми | Кількість годин |
|-------|--|-----------------|
| 1 | Основи роботи в середовищі SWI Prolog. Складання Пролог-програми | 4 |
| 2 | Арифметичні операції. Метод рекурсії. Робота зі списками | 6 |
| 3 | Бінарні дерева | 5 |
| 4 | Довизначення елементарних вбудованих функцій мови Лісп. Побудова найпростіших функцій. | 4 |
| 5 | Побудова та відлагодження рекурсивної функції з підфункціями. | 6 |
| 6 | Побудова та відлагодження програм символічного диференціювання математичних виразів з реалізацією основних операцій. | 5 |
| | Разом | 30 |

6.4. Самостійна робота

| № з/п | Назва теми | Кількість годин |
|-------|---|-----------------|
| 1 | Основні елементи мови Пролог. | 8 |
| 2 | Рекурсивне представлення даних і програм у Пролог. | 12 |
| 3 | Технології побудови експертних систем. | 10 |
| 4 | Побудова рекурсивних функцій у Лісп | 7 |
| 5 | Спискова форма представлення даних у функціональному програмуванні. | 7 |
| 6 | Суть крапкової форми подання виразів | 6 |
| 7 | Додаткові можливості функціональної мови Лісп. | 10 |
| | Разом | 60 |

7. ІНСТРУМЕНТИ, ОБЛАДНАННЯ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ВИКОРИСТАННЯ ЯКИХ ПЕРЕДБАЧАЄ НАВЧАЛЬНА ДИСЦИПЛІНА

Лабораторні роботи виконуються на персональних комп'ютерах із встановленою операційною системою Windows, Linux. Програмне забезпечення: SWI Prolog, Lisp.

8. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Заяць В.М., Заєць М.М., Гордукова І.Є. Логічне функціональне програмування: навч. посіб. Львів: Нац. ун-т "Львівська політехніка". 2016. – 398 с.
2. Логічне і функціональне програмування. Системний підхід. Підручник / Заяць В.М., Заяць М.М. – Рівне: НУВГП, 2018. – 422 с.
3. Шекета В.І. Логічне програмування: метод. вказівки / В. І. Шекета. – Івано-Франківськ: ІФНТУНГ, 2009. – 91 с.
4. Юрчишин В.М. Пролог – мова логічного програмування: конспект лекцій / В. М. Юрчишин, В. І. Шекета, Л. М. Гобир. – Івано-Франківськ: ІФНТУНГ, 2005. – 138 с.
5. Eduardo Costa. Visual Prolog 7.3 for Tyros. / Eduardo Costa. – New York: Springer-Verlag, 2010. – 270 p.

Інформаційні ресурси:

6. Електронний науковий архів НУ «Львівська політехніка» / [Електронний ресурс] – Режим доступу: <http://ena.lp.edu.ua>
7. Національна бібліотека України імені В.І. Вернацького [Електронний ресурс] Режим доступу: http://www.irbis-nbuv.gov.ua/cgi-bin/irbis64r_81/
8. Цифрова бібліотека факультету електроніки НТТУ «КПІ» / [Електронний ресурс] – Режим доступу: <http://fel.kpi.ua/>
9. Електронний науковий архів НУ «Львівська політехніка» / [Електронний ресурс] – Режим доступу: <http://ena.lp.edu.ua>

ДОДАТОК

ПЕРЕЛІК ПИТАНЬ І ЗАДАЧ ПЕРШОГО МОДУЛЬНОГО КОНТРОЛЮ

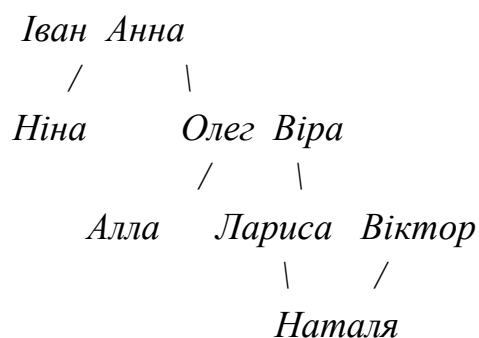
Питання

1. Основні поняття логічного програмування: поняття логічної програми. Поняття термінів та їх уніфікація; правила резолюцій.
2. Основні конструкції. Факти, правила, питання.
3. Структура програми на Пролозі.
4. Основні елементи мови: алфавіт мови. Терми, їх види: константи, змінні, структури. Літери і їхні типи. Інтерпретація літер.
5. Оператори. Властивості операторів (позиція, пріоритет, асоціативність). Інфіксні, префіксні, постфіксні оператори. Запис фактів і правил. Предикат.
6. Арифметика в Пролозі. Вбудовані предикати для порівняння чисел. Обчислення арифметичних виразів. Предикат is. Приклади програм з виконанням арифметичних операцій.

7. Побудова рекурсивних програм. Граничні умови і способи використання рекурсії.
8. Управління механізмом повернення. Відсікання і способи його використання. Причини використання відсікання.
9. Робота зі списками.
10. Бінарні дерева.
11. Предикати вводу й виводу. Приклади програм з використанням введення і висновку.
12. Створення динамічних баз даних: предикати для роботи з базою даних, створення бази даних, яка розташовується в оперативній пам'яті.
13. Технології побудови експертних систем. Експертні системи, що ґрунтуються на правилах. Експертні системи, що ґрунтуються на законах логіки.

Задачі

1. Описати засобами Пролога (за допомогою фактів) дерево родинних відносин, використовуючи предикат `roditel` з двома параметрами: ім'я батьків і ім'я дитини.
- 2.



У вікні діалогу сформувані наступні питання:

1. Чи є Іван батьком Ніни?
 2. Чи є Іван батьком Алли?
 3. Хто батьки Лариси?
 4. Як звуть дітей Олега?
 5. Хто батько батька Наталі?
 6. Хто чий батько?
 7. Чи є у Ніни і Олега спільний з батьків?
 8. Як звуть жінку Івана?
 9. Хто у Анни онуки?
 10. Чи є у Лариси брат або сестра?
2. На мові Пролог написати програму для знаходження більшого із 4 чисел.
 3. Просумувати цілі додатні числа, які слідує одне за одним з кроком `d` і закінчуються числом `n`. Значення `d` і `n` вводяться по запиту з екрана дисплея.

(Наприклад, $n=11$, $d=3$, сумуються числа $11+8+5+2=26$). В випадку $d>n$ сума дорівнює n .

4. Написати програму, яка знаходить найбільший спільний дільник двох чисел.
5. Написати програму, яка розв'язує квадратне рівняння виду:
$$Ax^2 + Bx + C = 0.$$
6. Обчислити суму ряду цілих парних та непарних чисел від 1 до n .
7. Написати програму сортування списку бульбашковим методом, результатом роботи програми повинен бути вхідний і відсортований список.
8. Додати заданий елемент в кінець списку.
9. Написати програму знаходження n -го члену ряду Фібоначчі та суму перших n членів цього ряду.
10. Із заданого списку утворити полідром. Полідром - це список, який читається однаково як зліва направо, так і в протилежному напрямку. Приклад полідрома – $[1,2,3,2,1]$.
11. Написати програму сортування списку методом вставки, результатом роботи програми повинен бути вхідний і відсортований список.
12. Описати предикат пошуку максимального елемента бінарного дерева. Вивести максимальний елемент та бінарне дерево.
13. Описати предикат підрахунку листків бінарного дерева та їх виведенню.

ПЕРЕЛІК ПИТАНЬ І ЗАДАЧ ДРУГОГО МОДУЛЬНОГО КОНТРОЛЮ

Питання

1. Історія розвитку функціонального програмування. Властивості і переваги функціонального програмування.
2. Основи функціонального програмування. Порівняння можливостей мов функціонального та логічного програмування з іншими мовами високого рівня. Лямбда обчислення. Величини. Функції. Програмування за допомогою функцій та процедур.
3. Порівняння можливостей мов функціонального та логічного програмування з іншими мовами високого рівня. Функціональні мови програмування.
4. Строго функційна мова. Елементарні поняття.
5. Символьні дані: атоми, константи, списки, символьні вирази та представлення даних.
6. Побудова рекурсивних функцій: означення рекурсії, розгляд випадків, вибір підфункцій, підходи до побудови, запис S- виразу.
7. Спискова форма представлення даних у функціональному програмуванні. Принципи опрацювання списків.
8. Вкладання списків на довільну глибину. Спрощена форма запису.

9. Суть крапкової форми подання виразів. Використання його до побудови функціональних програм. Приклади побудови функціональних програм.
10. Принципи структурованої побудови програм та їх структурованого відлагоджування.
11. Побудова програм аналізу розмірностей математичних формул – як приклад структурованої програми. Два підходи до подання розмірностей.
12. Додаткові можливості функціональної мови Лісп.
13. Функції вищих порядків. Локальні форми, лямбда-вирази, параметри нагромадження та їх застосування до побудови функціональних програм.
14. Означення та побудова функцій вищих порядків, їх функціональне означення та запис S-виразів.

Задачі

1. Визначити наступні функції:
 - Функція `max3` по трьох цілих числах визначає найбільше значення із них.
 - Функція `sort2`, по двох цілих повертає пару, в якій найменше із них стоїть на першому місці, а найбільше — на другому.
 - Функція `bothTrue`, яка повертає `true` тоді і тільки тоді, коли обидва її аргументи будуть рівні `true`.
 - Функція `solve2`, яка по двох числах, які представляють собою коефіцієнти лінійного рівняння $ax + b = 0$, повертає пару, перший елемент якої рівний `true`, якщо розв'язок існує і `false` в протилежному випадку; при цьому другий елемент дорівнює або значенню кореня, або `0.0`.
 - Функція `isTriangle`, яка визначає, чи можна із відрізків з заданими довжинами x , y і z побудувати трикутник.
2. Визначити функцію, що має вхідними даними ціле число n і визначаючу список, що містить n елементів, впорядкованих по зростанню.
 - список парних натуральних чисел.
 - список факторіалів.
3. Визначити наступні функції:
 - Функція додавання елементів двох списків. Повертає список, складений із сум елементів списків-параметрів. Врахувати, що передані списки можуть бути різної довжини.
 - Функція `removeOdd`, яка удаляє із заданого списку цілих чисел всі непарні числа. Наприклад: `removeOdd [1, 4, 5, 6, 10]` повинен повертати `[4, 10]`.
4. Визначіть запис, який представляє інформацію про карту в картовій грі. Кожна карта характеризується однією із чотирьох мастей. Карта може бути або молодшою (від двійки до десятки), або картинкою (валет, дама, король, туз).
Визначте наступні функції:
 - 1) Функція `isMinor`, яка перевіряє, чи є її аргумент молодшою картою.
 - 2) Функція `sameSuit`, яка перевіряє, чи передані до неї карти є однієї масті.

- 3) Функція `beats`, яка перевіряє, чи карта, передана їй як перший аргумент, б'є карту, яка є другим аргументом.
- 4) Функція `beats2`, аналогічна `beats`, але приймає в якості додаткового аргументу козирну масть.
- 5) Функція `beatsList`, яка приймає як аргумент список карт, карту й козирну масть і повертає список тих карт із першого аргументу, які б'ють зазначену карту з врахуванням козирної масті.
- 6) Функція, яка по заданому списку карт повертає список чисел, кожне з яких є можливою сумою очок зазначених карт, розрахованих за правилами гри в «двадцять одно»: молодші карти вважаються по номіналу, валет, дама й король вважаються за 10 очок, туз може розглядатися і як 1 і як 11 очок. Функція повинна повернути всі можливі варіанти.
5. Задано три процеси A, B і C. У кожного з них є локальний масив цілих чисел. Всі три масиви відсортовані у не спадаючому порядку і містять хоча б один елемент. Написати програму в якій всі три процеси взаємодіють, щоб знайти найменше спільне число.
6. В кімнату заходять процеси двох типів A і B. Процес типу A не може вийти поки не зустріне два процеси типу B, а B не може вийти поки не зустріне один процес типу A. Як тільки процес зустріне необхідну кількість процесів іншого типу він виходить із кімнати. Розробити серверний процес для реалізації такої синхронізації.
7. Розробити програму, яка здійснює множення матриць, причому обчислення кожного рядка нової матриці виконується окремим процесом.